Algorithm Design

M. T. Goodrich and R. Tamassia

John Wiley & Sons

Solution of Exercise C-10.7

Let $p(n)$ and $q(n)$ be the two polynomials, expressed by their coefficients. Rewrite these as $p(n) = p_1(n)x^{n/2} + p_2(n)$ and $q(n) = q_1(n)x^{n/2} + q_2(n)$. Let $r(n) = p_1(n) - p_2(n)$ and let $s(n) = q_2(n) - q_1(n)$, and note that $r(n)s(n) = p_1(n)q_2(n) - p_2(n)q_2(n) - p_1(n)q_1(n) + p_1(n)q_2(n)$. Thus, $r(n)s(n) + p_1(n)q_1(n) + p_2(n) + q_2(n) = p_1(n)q_2(n) + p_1(n)q_2(n)$. Note that $p(n)q(n) = p_1(n)q_1(n)x^n + [p_1(n)q_2(n) + p_1(n)q_2(n)]x^{n/2} + p_2(n)p_2(n)$, which is the same as $p_1(n)q_1(n)x^n + [r(n)s(n) + p_1(n)q_1(n) + p_2(n) + q_2(n)]x^{n/2} + p_2(n)p_2(n)$. Therefore, with three recursive calls on problems of size $n/2$, plus an additional amount of work that is $O(n)$, we can multiply $p(n)$ and $q(n)$. This gives us the recurrence $T(n) = 3T(n/2) + bn$, for some contant b, which implies that $T(n)$ is $O(n^{\log_2 3})$, by the master theorem.